

# Team 2 Pop Task

## Paul Wei, Yuanqin Fan, Chia-Ying Lin

Product Owner: Chia-Ying Lin(chiayinl)

### Product backlog

✓	WD-7	Create tasks	DOCUMENTATION	1
✓	WD-12	UI wireframe	DOCUMENTATION	1
✓	WD-33	Choose UI Frameworks	DOCUMENTATION	1
✓	WD-34	Research in New Backend Technologies	DOCUMENTATION	2
✓	WD-23	Login/ Logout/ Register Backend	USER ACCOUNT MANAGEMENT	1
✓	WD-17	Login/Register UI implementation	USER ACCOUNT MANAGEMENT	1
✓	WD-24	Task Model/ Form Creation	PERSONAL TASK CALENDAR VIEW	1
✓	WD-19	Task Add/Modify UI implementation	PERSONAL TASK CALENDAR VIEW	2
✓	WD-31	Task Add/ Modify Backend	PERSONAL TASK CALENDAR VIEW	3
✓	WD-35	Group Model Creation	PERSONAL TASK CALENDAR VIEW	1
✓	WD-25	Role Model Creation	PERSONAL TASK CALENDAR VIEW	1
✓	WD-37	UI Implementation for Team Management	PERSONAL TASK CALENDAR VIEW	2
✓	WD-36	Managing Group members backend	PERSONAL TASK CALENDAR VIEW	1
✓	WD-15	Pool UI implementation	TASK POOL	3
✓	WD-27	Task Pool Backend	TASK POOL	2
✓	WD-14	Calendar UI implementation	PERSONAL TASK CALENDAR VIEW	3
✓	WD-29	Calendar Backend	PERSONAL TASK CALENDAR VIEW	2
✓	WD-11	Sprint1 slides	DOCUMENTATION	1

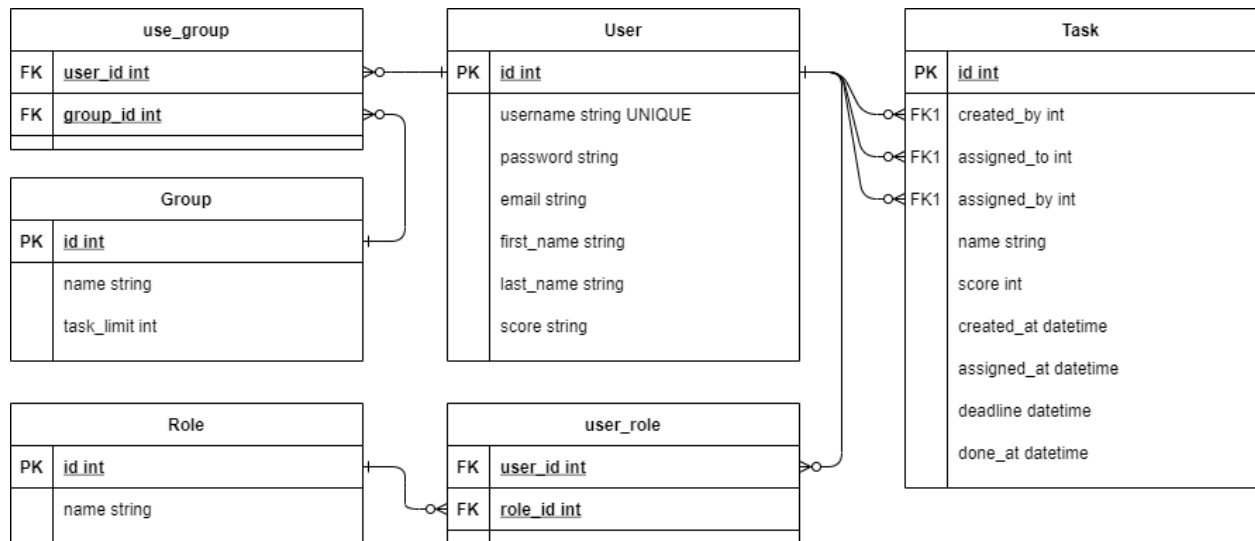
✓	WD-16	Leaderboard UI implementation	LEADER BOARD	2
✓	WD-26	Leader Board Backend	LEADER BOARD	2
✓	WD-18	Navigation bar UI implementation	PERSONAL TASK CALENDAR VIEW	2
✓	WD-28	Customize calendar UI	PERSONAL TASK CALENDAR VIEW	2
✓	WD-30	Supporting Task Item Dragging in Calendar	PERSONAL TASK CALENDAR VIEW	1
✓	WD-32	Supporting Task Item Dragging between Pool and Calendar	PERSONAL TASK CALENDAR VIEW	1
✓	WD-39	Task Complete Backend	TASK POOL	2
✓	WD-40	SSL/HTTPS implementation		2
✓	WD-41	User data sanitization		1
✓	WD-42	OWASP security check		2
✓	WD-21	Security Test		3
✓	WD-20	Deploy to AWS		3
✓	WD-22	Beta Test		2
✓	WD-43	Website load test		1
✓	WD-13	Sprint2 slides	DOCUMENTATION	1

## Sprint #1 backlog

✓ WD-7 Create tasks	DOCUMENTATION	1
✓ WD-12 UI wireframe	DOCUMENTATION	1
✓ WD-33 Choose UI Frameworks	DOCUMENTATION	1
✓ WD-34 Research in New Backend Technologies	DOCUMENTATION	2
✓ WD-23 Login/ Logout/ Register Backend	USER ACCOUNT MANAGEMENT	1
✓ WD-17 Login/Register UI implementation	USER ACCOUNT MANAGEMENT	1
✓ WD-24 Task Model/ Form Creation	PERSONAL TASK CALENDAR VIEW	1
✓ WD-19 Task Add/Modify UI implementation	PERSONAL TASK CALENDAR VIEW	2
✓ WD-31 Task Add/ Modify Backend	PERSONAL TASK CALENDAR VIEW	3
✓ WD-35 Group Model Creation	PERSONAL TASK CALENDAR VIEW	1
✓ WD-25 Role Model Creation	PERSONAL TASK CALENDAR VIEW	1
✓ WD-37 UI Implementation for Team Management	PERSONAL TASK CALENDAR VIEW	2
✓ WD-36 Managing Group members backend	PERSONAL TASK CALENDAR VIEW	1
✓ WD-15 Pool UI implementation	TASK POOL	3
✓ WD-27 Task Pool Backend	TASK POOL	2
✓ WD-14 Calendar UI implementation	PERSONAL TASK CALENDAR VIEW	3
✓ WD-29 Calendar Backend	PERSONAL TASK CALENDAR VIEW	2
✓ WD-11 Sprint1 slides	DOCUMENTATION	1

# Data model

## Object model



The PopTask system has the following entities:

- User
- Task: minimal item that user can create and take as to-do item
- Group: one or more users in which share public task pool
- Role: define access to different functionalities; e.g. a manager can assign task to other users in a group

Relations:

- User can have one or more tasks, while a task can only be assigned to one user
- User can be in one or more groups
- User can have one or more roles

## Django Model Implementation

```
class Profile(models.Model):
    score = models.IntegerField(default=0)
    picture = models.FileField(blank=True)
    content_type = models.CharField(max_length=50)
    user = models.OneToOneField(User, on_delete=models.PROTECT,
related_name="profile")

class Task(models.Model):
    created_by = models.ForeignKey(User, on_delete=models.PROTECT,
related_name="tasks_created")
```

```

        assigned_by = models.ForeignKey(User, on_delete=models.PROTECT,
related_name="tasks_giveout")
        assigned_to = models.ForeignKey(User, on_delete=models.PROTECT,
related_name="tasks_assigned")
        name = models.CharField(max_length=200)
        score = models.IntegerField(
            default=1,
            validators=[
                MinValueValidator(1)
            ])
        creation_at = models.DateTimeField()
        assigned_at = models.DateTimeField()
        deadline = models.DateTimeField()
        done_at = models.DateTimeField()
        class Meta:
            ordering = ('deadline', 'score')

class Group(models.Model):
    name = models.CharField(max_length=200)
    task_limit = models.IntegerField(
        default=1,
        validators=[
            MinValueValidator(1)
        ])
    members = models.ManyToManyField(User, related_name="mygroups")

class Role(models.Model):
    name = models.CharField(max_length=200)
    assigned_to = models.ManyToManyField(User, related_name="roles")


```

# Project Wireframe

## PopTask





A web tool to help people prioritize and manage task deadline easily

## Home Page (Login / Create Account)


 Pop Task


Task Sapce


Leader Board





Web Dev Team 2


Team Members 

 Member 1

 Member 2

 Member 3

 Member 4

 Member 5


Task Pool

☐

Task View

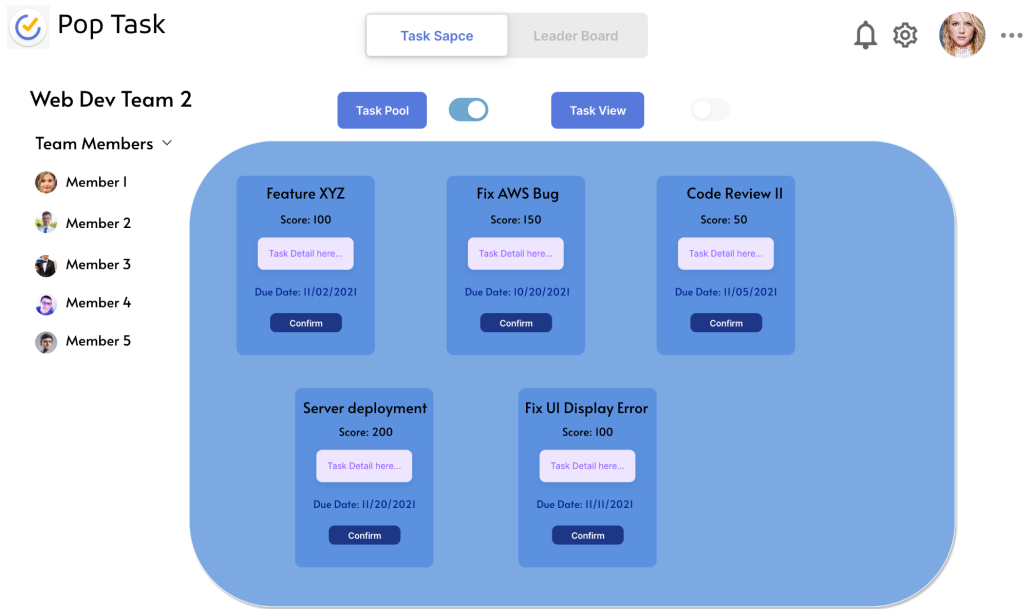
☒

October 2021

Mon	TU	WED	THU	FRI	Weekend
4	5	6	7	8	
11	12	13	14	15	
18  Fix AWS Bug	19	20	21 Feature XYZ	22	
25	26 Fix UI Display Error	27	28	29	

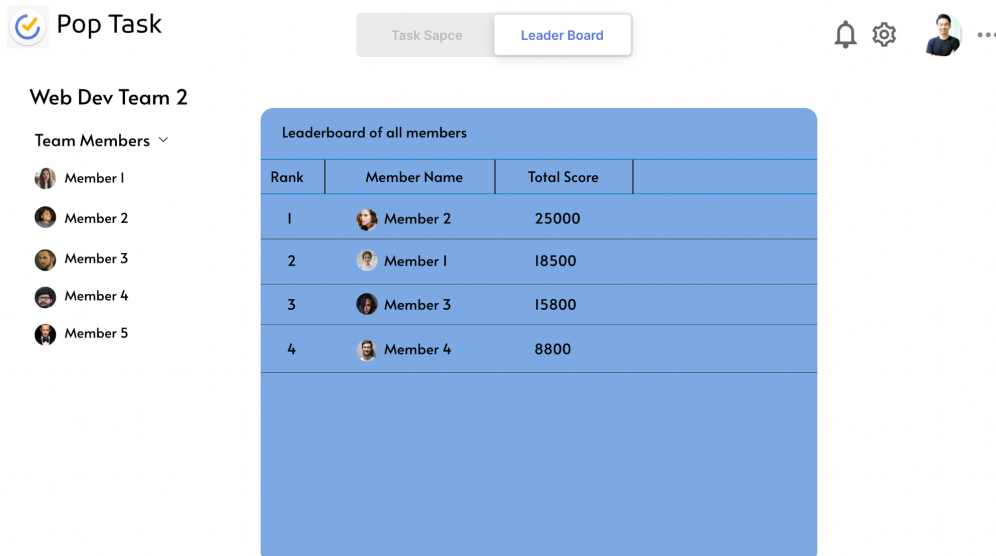
## Task Space Page (task view)

- Team information(Team member, Team name, Team setting)
- Task information in calendar view
  - task details and description, burndown bar to show the priority of each task)



### Task Space Page(task pool)

- All of the pending task information(score, due date and detail) in board view
- Users can use “confirm button” to claim the task they want and add it to their task view



### Leaderboard Page

- All of the team members in the same group will be sorted by their score and listed in the scoreboard.